ROLL NO: 210701268

Exp5a: Design and test various schema models to optimize data storage and retrieval Using Hive.

Aim:

To Design and test various schema models to optimize data storage and retrieval Using Hbase.

Procedure:

Step 1: Start Hive

Open a terminal and start Hive by running:

$hive

Step 2: Create a Database

Create a new database in Hive: hive>CREATE DATABASE financials;

```
hive> CREATE DATABASE financials;
OK
Time taken: 0.063 seconds
```

***Step 3: Use the Database**:*

Switch to the newly created database: hive>use financials;

```
hive> use financials;
OK
Time taken: 0.066 seconds
```

Step 4: Create a Table:

Create a simple table in your database:

hive>CREATE TABLE finance_table(id INT, name STRING);

```
hive> CREATE TABLE finance_table (
    > id INT,
    > name STRING
    > );
OK
Time taken: 0.768 seconds
```

Step 5: Load Sample Data:

You can insert sample data into the table:

hive>INSERT INTO finance_tableVALUES (1, 'Alice'), (2, 'Bob'), (3, 'Charlie');

```
hive> INSERT INTO finance_table VALUES
    >  (1, 'Alice'),
    >   (2, 'Bob'),
    >   (3, 'Charlie');
Query ID = hadoop_20231028192937_fdebeb4e-abf7-4bad-a248-ac908246e3c1
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2023-10-28 19:29:41,158 Stage-1 map = 0%,  reduce = 0%
```

Step 6: Query Your Data

Use SQL-like queries to retrieve data from your table:

hive>CREATE VIEW myview AS SELECT name, id FROM finance_table;

Step 7: View the data:

*To see the data in the view, you would need to query the view hive>SELECT*FROM myview;*

```
hive> SELECT * FROM myview;
OK
Alice   1
Bob     2
Charlie 3
Time taken: 0.238 seconds, Fetched: 3 row(s)
```

Step 8: Describe a Table:

You can describe the structure of a table using the DESCRIBE command:

hive>DESCRIBE finance_table;

```
hive> DESCRIBE finance_table;
OK
id                      int
name                    string
Time taken: 0.081 seconds, Fetched: 2 row(s)
```

Step 9: Alter a Table:

You can alter the table structure by adding a new column: hive>ALTER TABLE finance_table ADD COLUMNS (age INT);

```
hive> ALTER TABLE finance_table ADD COLUMNS (age INT);
OK
Time taken: 0.165 seconds
```

Step 10: Quit Hive:

To exit the Hive CLI, simply type: hive>quit;

```
sudhashreem@sudhashreem-VirtualBox:~/DA/exp4$ cd ..
sudhashreem@sudhashreem-VirtualBox:~/DA$ cd ..
sudhashreem@sudhashreem-VirtualBox:~$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/sudhashreem/hive/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/sudhashreem/hadoop/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticL
oggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = f658134c-3a88-4e89-b5c3-a1de3c652802

Logging initialized using configuration in jar:file:/home/sudhashreem/hive/lib/hive-common-3.1.2.jar!/hive-log4j2.properties Async:
true
Hive Session ID = 88cb5f56-7fc1-44e5-a5f6-2c6a09794be2
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e
. spark, tez) or using Hive 1.X releases.
hive> CREATE DATABASE financials;
OK
Time taken: 0.954 seconds
hive> use financials;
OK
Time taken: 0.185 seconds
hive> CREATE TABLE finance_model( id INT, name STRING);
OK
Time taken: 1.0 seconds
hive> INSERT INTO finance_model VALUES (1,'Alice'),(2,'Bob'),(3,'Candice');
Query ID = sudhashreem_20240917160510_56998f35-1574-4484-9b8f-3ea7049d4d70
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1726567709224_0004, Tracking URL = http://sudhashreem-VirtualBox:8088/proxy/application_1726567709224_0004/
Kill Command = /home/sudhashreem/hadoop/bin/mapred job  -kill job_1726567709224_0004
```

```
Kill Command = /home/sudhashreem/hadoop/bin/mapred job  -kill job_1726567709224_0004
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2024-09-17 16:05:29,876 Stage-1 map = 0%,  reduce = 0%
2024-09-17 16:05:42,577 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 7.31 sec
2024-09-17 16:05:50,284 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 9.74 sec
MapReduce Total cumulative CPU time: 9 seconds 740 msec
Ended Job = job_1726567709224_0004
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://localhost:9000/user/hive/warehouse/financials.db/finance_model/.hive-staging_hive_2024-09-17_16-05-1
0_478_3670786972070531277-1/-ext-10000
Loading data to table financials.finance_model
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 9.74 sec   HDFS Read: 15721 HDFS Write: 291 SUCCESS
Total MapReduce CPU Time Spent: 9 seconds 740 msec
OK
Time taken: 45.167 seconds
hive> CREATE VIEW myview AS SELECT name, id FROM  finance_model;
OK
Time taken: 0.516 seconds
hive> SELECT * FROM myview;
OK
Alice   1
Bob     2
Candice 3
Time taken: 0.496 seconds, Fetched: 3 row(s)
hive> DESCRIBE finance_model;
OK
id                      int
name                    string
Time taken: 0.167 seconds, Fetched: 2 row(s)
hive> ALTER TABLE finance_model ADD COLUMNS (age INT);
OK
Time taken: 0.476 seconds
hive> quit;
sudhashreem@sudhashreem-VirtualBox:~$
```

Result:

Thus, the usage of various commands in Hive has been successfully completed.